# Scheduling analysis of AADL architecture models

**Frank Singhoff+, Pierre Dissaux***

**+Lab-STICC/CNRS UMR 6285, Université de Bretagne Occidentale, France**
***Ellidiss Technologies, France**

# Outline

**Goal:** overview of scheduling analysis capabilities that are proposed by the AADL and tools implementing it. Show the benefits that can be expected by performing early scheduling analysis for real-time software.

- **Part 1: introduction to AADLv2 core (about 2h/2h30)**
  - Syntax, semantics of the language
- **Part 2: introducing a case study (about 15')**
  - A radar illustrative case study
- **Part 3: scheduling analysis (about 2h/2h30)**
  - Introducing real-time scheduling and its use with AADL
- **Part 4: practical labs, exercises, discussion (about 1 or 2 hours)**
  - How to use tools in order to apply what we learnt in parts 1 to 3

# CPS-WEEK Agenda

- 9:00-10:00 tutorial
- 10:00-10:30 coffee break
- 10:30-12:30 tutorial
- 12:00-13:30 lunch break
- 14:00-15:00 tutorial
- 15:00-15:30 coffee break
- 15:30-17:30 tutorial

# Acknowledgments

- Many of those slides were written with or by Jérôme Hugues/ISAE, for the following tutorials:
  - AADLv2, An Architecture Description Language for the Analysis and Generation of Embedded Systems. J. Hugues, F. Singhoff. Half day tutorial presented in the ACM HILT conference, Portland, USA, October 2014.
  - AADLv2, a Domain Specific Language for the Modeling, the Analysis and the Generation of Real-Time Embedded Systems. F. Singhoff, J. Hugues. Half day tutorial presented in the International MODELS conferences, Valencia, Spain, September 2014.
  - AADLv2, an Architecture Description Language for the Analysis and Generation of Embedded Systems. J. Hugues F. Singhoff. Half day tutorial presented in the International EMSOFT/ESWEEK conferences, Montreal, Canada, September 2013.
  - Développement de systèmes à l'aide d'AADL - Ocarina/Cheddar. J. Hugues, F. Singhoff. Tutoriel présenté à l'école d'été temps réel (ETR'2009). Septembre 2009. Pages 25-34. Paris.
- Thank you Jérôme :-)

# We focus on Real-Time, Critical, Embedded Systems

- *« The correctness of the system depends not only on the logical result of computation, but also on the time at which the results are produced »* Stankovic, 1988.

- **Properties we look for:**
  - Functions must be predictable: the same data input will produce the same data output.
  - Timing behavior must be predictable: must meet temporal constraints (e.g. deadline).

# We focus on Real-Time, Critical, Embedded Systems

- **Critical real-time systems:** temporal constraints MUST be met, otherwise defects could have a dramatic impact on human life, on the environment, on the system,

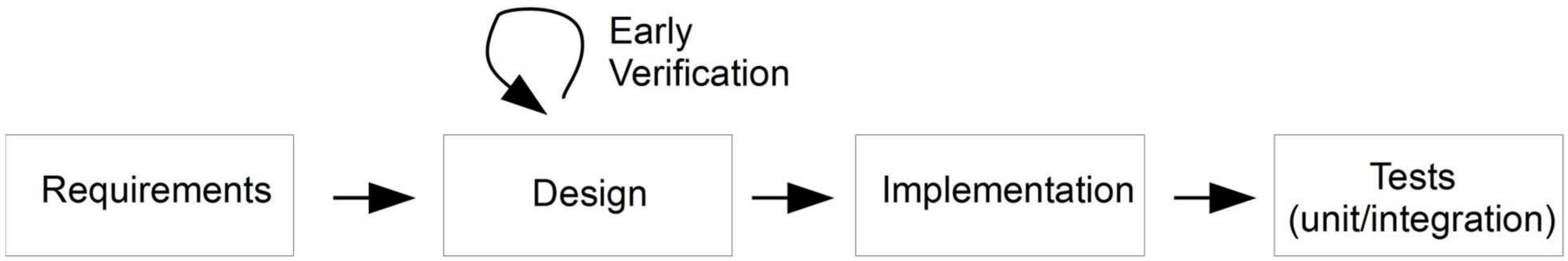
- **Embedded systems:** computing system designed for specific control functions within a larger system.
  - Often with temporal constraints.
  - Part of a complete device, often including hardware and mechanical parts
  - Limited amount of resources.

# We focus on Real-Time, Critical, Embedded Systems

- **Examples:** aircraft, satellite, automotive, …

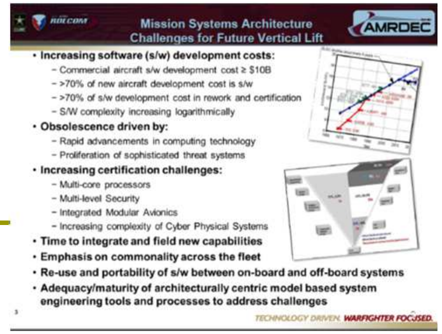
1. Need to handle time. Concurrent applications.
2. May have dramatic impact on human life, on the system, ...
3. Do not allow software maintenance => difficult to correct erroneous software/bugs.
4. High implementation cost : temporal constraints verification, safety, dedicated hardware/software

# We focus on Real-Time, Critical, Embedded Systems




- **Specific software engineering** methods/models/tools to master quality and cost
  - Example : early verifications at design step

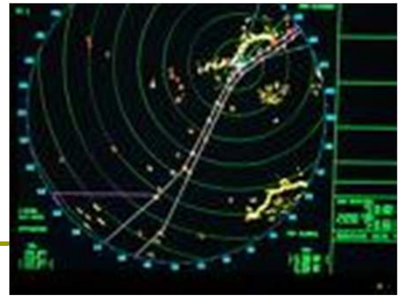
# Motivation for early verification




- **From NIST 2012:**
  - 70% of fault are introduced during the design step ; Only 3% are found/solved. Cost : x1
  - Unit test step: 20% of fault are introduced ; 16% are found/solved. Cost : x5
  - Integration test step: 10% of fault are introduced ; 50% are found/solved. Cost : x16

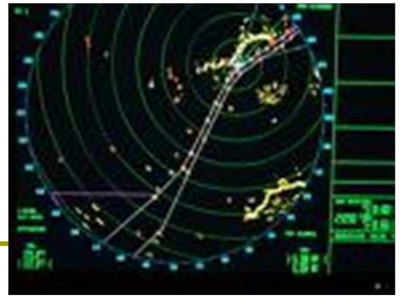
- **Objective:** increase the number of faults found at design step!
- **Early verification:** multiple verifications, including expected performances, e.g. can deadlines be met?

# Objectives of this tutorial



- Issues
  - How to model/design a real-time critical embedded system that conforms to requirements?
  - How to verify the solution?
  - How to simulate it?
  - How to implement it (not in this tutorial!)?
- One solution amoung others:  use an architecture description language
    - to model the system,
    - to run various verification,
    - and to automatically produce the system
- Focus on the AADL2.2 SAE standard

# Objectives of this tutorial



- Illustration: model of a simple radar system
- Let us suppose we have the following requirements

1. System implementation is composed by physical devices (Hardware entity): antenna + processor + memory + bus
2. and software entities : running processes and threads + operating system functionalities (scheduling) implemented in the processor that represent a part of execution platform and physical devices in the same time.
3. The main process is responsible for signals processing : general pattern: transmitter -> antenna -> receiver -> analyzer -> display
4. Analyzer is a periodic thread that compares transmitted and received signals to perform detection, localization and identification.
5. [..]

# Resources for this tutorial

- Information on AADL
  - http://www.aadl.info : updates on AADL standard
  - http://www.openaadl.org : many AADL resources
  - http://www.ellidiss.fr/: AADLInspector and Ellidiss Tech. AADL activities
  - http://beru.univ-brest.fr/~singhoff/cheddar/: Cheddar and real-time scheduling

- Feel free to contact us for more details

# Outline

**Goal:** overview of scheduling analysis capabilities that are proposed by the AADL and tools implementing it. Show the benefits that can be expected by performing early scheduling analysis for real-time software.

- **Part 1: introduction to AADLv2 core (about 2h/2h30)**
  - Syntax, semantics of the language
- **Part 2: introducing a case study (about 15')**
  - A radar illustrative case study
- **Part 3: scheduling analysis (about 2h/2h30)**
  - Introducing real-time scheduling and its use with AADL
- **Part 4: practical labs, exercises, discussion (about 1 or 2 hours)**
  - How to use tools in order to apply what we learnt in parts 1 to 3